A REPORT

ON

AADHAAR MASKING TOOL

BY

NACHIKETSINGH kANDARI 2021A7PS2691P

AT

FUTURE GENERALI INDIA LIFE INSURANCE COMPANY LIMITED,

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

JULY 2023

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)
Practice School Division

Station: Future Generali India Life Insurance Company Limited
Duration: 2 MONTHS
Date of Start: 29/05/23
Date of Submission: 21/07/23

.....

Title of the Project: AADHAR MASKING TOOL

Name of Student:
Nachiketsingh Kandari - 2021A7PS2691P

Name of Expert(s): Paritosh Darekar, Nikita Malviya

Name of PS instructor: Dr. Saurabh Chandrakant Patankar

Key Words: Optical Character Recognition, ML, Computer Vision

Project Areas: Computer Vision on handwritten applications.

Signature:

INDEX

ACKNOWLEDGEMENTS----------Page 3

ABSTRACT ----------Page 4

INTRODUCTION ----------Page 5

BACKGROUND ----------Page 6
- Technologies Used---------- Page 6
- Reference Discussion---------- Page 6
- Alternative Solutions----------Page 7
- Shortcomings Of The Alternatives----------Page 7
- Advantages Of My Project----------Page 7

PROJECT ----------Page 8
- Project Roadmap ----------Page 8
- Project Phases ---------- Page 8
- Comparison of Solutions----------Page 17

CONCLUSION----------Page 19
- Accomplishments and Learnings ----------Page 19
- Future Work----------Page 19
- Self - Review ----------Page 19

REFERENCES ---------- Page 20

ACKNOWLEDGEMENTS

I am writing to express my heartfelt gratitude for the invaluable learning experience I gained during my internship at Future Generali India Life Insurance. I am extremely thankful for the opportunity to be a part of your esteemed organization and to contribute to its goals and objectives.

I would like to acknowledge and express my deepest appreciation to everyone who supported me throughout my internship, including DR. SAURABH CHANDRAKANT PATANKAR. I am grateful for his guidance, expertise, and continuous support throughout my internship. Your valuable insights and constructive feedback played a crucial role in shaping my understanding.

I am grateful to my mentor, PARITOSH DAREKAR, for choosing this particular project and entrusting me with it. His faith in me along with his constant support and feedback were a source of inspiration for me to keep improving my project to meet the industry standards. It is due to his guidance that I could experience the entire development process of an application.

I would also like to express my gratitude towards my mentor NIKITA MALVIYA for taking the time to guide me, answer my questions, and provide valuable insights to develop a UI solution which best captures the look and feel of Future Generali. Your expertise and willingness to share your knowledge were instrumental in shaping my professional growth during this internship.

Lastly, I would like to acknowledge the Practice School Division of my college, BITS Pilani. Their continuous efforts in organizing and facilitating this practice school program have provided me with a platform to apply my theoretical knowledge in a real-world setting. I am grateful for their commitment to hands-on learning, which has significantly contributed to my overall development . I am also attaching my PS report for your perusal. It encompasses the knowledge and skills I acquired during my PS-1 and reflects my understanding of the assigned tasks and projects.

ABSTRACT

The RBI has issued a mandate which requires the Aadhaar details to be masked in all the KYC (know your customer) documents that a company stores on their database.
Aadhaar masking refers to blacking out of the digits of the Aadhaar number in a saved Aadhaar document. This is done in order to keep Aadhaar data private and secure.

Future Generali, being an insurance company, had many scanned versions of handwritten applications in their database which needed to have their Aadhaar number masked.
There existed a solution within Future Generali to mask the Aadhaar documents but it was of a generalized nature and hence it had low accuracy and speed. Furthermore, there exist subscription based API solutions for this but they have a cost factor and Future Generali wanted an in-house solution. Thus the masking process for existing documents was being done manually so far.

I was tasked to develop an Aadhaar Masking Tool to mask a particular type of scanned handwritten application forms. The project utilizes techniques in computer vision, image processing, and optical character recognition (OCR) to accurately detect Aadhaar card details from the scanned handwritten applications.

This project report goes through the process of developing a solution which showed an improvement of 550% in accuracy of masking (15% to 100%) and an improvement of 1600% in average time taken to mask a document (39s per document to 2.6 seconds per document).

The outcomes of this project provides the company a ready-to-deploy solution which can be easily integrated within their current system and which makes them compliant with the RBI mandate and saves time and effort that was otherwise being put in the task of manually masking the Aadhaar number from each document.

INTRODUCTION

The Reserve Bank of India issued a circular which mandates all regulated entities to mask the Aadhaar numbers of their customers from the Aadhaar image as part of the KYC process.

Future Generali being one of them, has to comply and hence I was tasked to develop an Aadhaar Masking Tool specifically designed to mask a certain category of application forms in order to ensure compliance with the RBI mandate.
These forms are currently being masked manually by the employees of the company.

Aadhaar masking, or masking of the Aadhaar numbers is an important step taken by RBI to ensure data privacy and security of the citizens. The core code can be tweaked to suit any such organizations' need to mask Aadhaar numbers from their applications.

In this report, I will talk about my approach to the problem and how I improved upon the existing solution to create a much faster and more accurate solution.

In the Background section, I will go through the technologies used, alternatives solutions that exist and their shortcomings and how my solution makes a difference.

BACKGROUND

Technologies Used

Pillow (PIL) - The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation. This library was used for basic opening, closing and image manipulation.

OpenCV - OpenCV stands for Open-Source Computer Vision (Library). It is the most common and popularly used, well-documented Computer Vision library. OpenCV is an open-source library that incorporates numerous computer vision algorithms. OpenCV increases computational efficiency and assists with real-time applications. For this project, OpenCV was extensively used to detect the bounding boxes before I could OCR them.

Pytesseract - Pytesseract or Python-tesseract is an Optical Character Recognition (OCR) tool for Python. It will read and recognize the text in images, license plates etc. Python-tesseract is actually a wrapper class or a package for Google's Tesseract-OCR Engine.

Streamlit - Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers.

Reference Discussion

OCR in Python : Playlist - This was the first recommendation from my mentor, Paritosh Darekar, and my introduction to the world of Optical Character Recognition. It is very educational and to the point, it provided me with most of the tools necessary for the completion of this project.

Handwritten Digit Recognition - This was the second recommendation from my mentor, Paritosh Darekar, and it introduced me to convolutional neural networks and model training. I did not use it in this project but it surely got added to my arsenal of OCR tools.

Streamlit Documentation - This is the official documentation for the Streamlit library within Python. I often found myself coming back here to enhance my knowledge on different features provided by Streamlit.

Stack Overflow - Stack Overflow is the largest, most trusted online community for developers to learn, share their programming knowledge and build their careers. This was my go-to for any doubts I had throughout the project. It helped me a lot to understand how to work with Streamlit and customize it to suit the company's needs.

Alternative Solutions

There are a plethora of Aadhaar Masking APIs and Software licensing companies which work on a subscription basis like IDfy, iSolve, aadhaarmasking.com, FRSLabs, etc. These work reliably well on Aadhaar card images. But these are not without their shortcomings. Even Future Generali itself had an Aadhar masking solution which was very good at detecting and masking Aadhaar as well as other document types.

Shortcomings Of The Alternatives

The alternatives present in the market have created a solution for the basic 3 forms of aadhar documents namely Aadhaar Letter, eAadhaar and Aadhaar PVC Card and do not currently have the capability to mask scanned handwritten applications that are submitted to the company. They are not customized for a particular application design and to find Aadhaar numbers in any document.

Advantages Of My Project

The lack of solutions which mask the data from handwritten application forms makes my project stand out since it introduces a tailored solution which works efficiently to mask the Aadhaar number on handwritten Future Generali applications. This in-house solution ensures data-security while saving outsourcing costs for the company at the same time.

PROJECT

To start off the project, I had to learn many new technologies and acquire new skills. It started off with learning how to OCR a scanned document and ended with the handwritten documents being successfully processed through a UI which I built. The project as an overview or a roadmap summarized in points looks like:

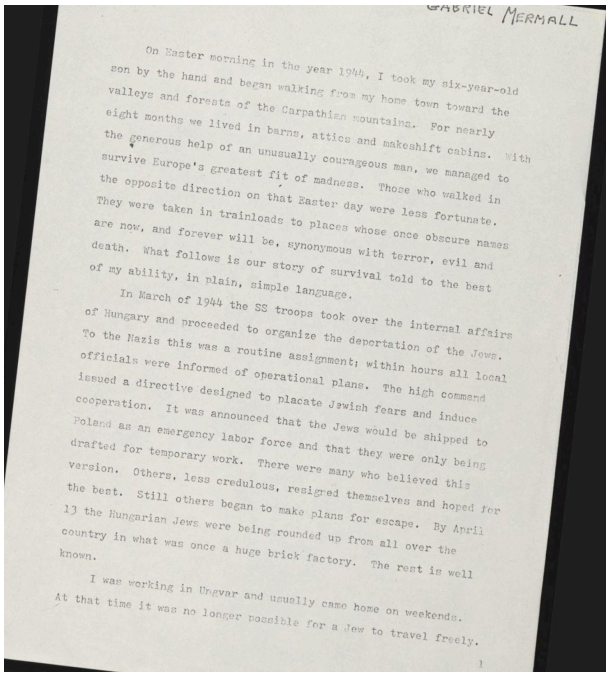
Project Roadmap

1. Gaining comprehensive knowledge of OCR techniques and algorithms.
2. Developing proficiency in image preprocessing to enhance OCR accuracy.
3. Implementing an algorithm to detect Aadhaar card details from scanned documents.
4. Masking sensitive information from the documents.
5. Optimizing the process to be fast and accurate.
6. Create a UI for the company to use this tool in their day-to-day operations.
7. Extra: Create a UI for the existing API solution.

Project Phases

OCR implementation on scanned documents

Embraced the challenge of starting from scratch and acquiring foundational knowledge of OCR techniques.
Explored libraries such as PILLOW, Pytesseract, and OpenCV for OCR implementation.
Successfully developed an OCR algorithm to extract text from scanned documents.

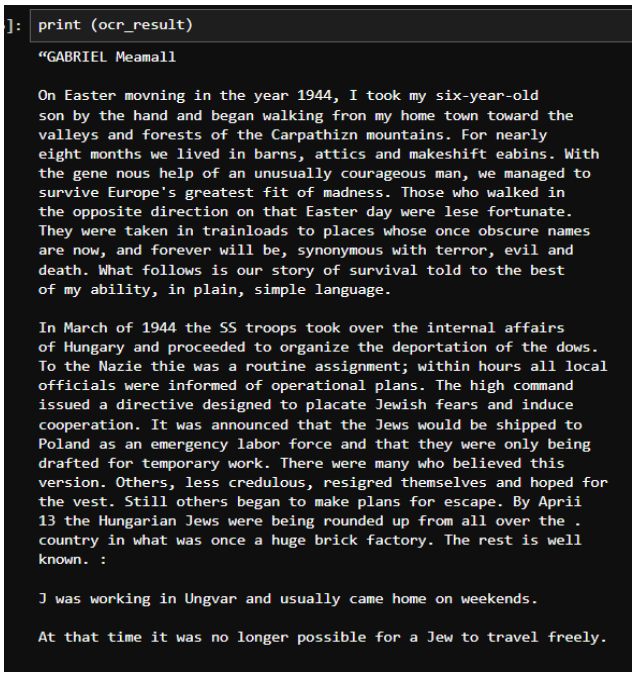


GABRIEL MERMALL

On Easter morning in the year 1944, I took my six-year-old son by the hand and began walking from my home town toward the valleys and forests of the Carpathian mountains. For nearly eight months we lived in barns, attics and makeshift cabins. With the generous help of an unusually courageous man, we managed to survive Europe's greatest fit of madness. Those who walked in the opposite direction on that Easter day were less fortunate. They were taken in trainloads to places whose once obscure names are now, and forever will be, synonymous with terror, evil and death. What follows is our story of survival told to the best of my ability, in plain, simple language.

In March of 1944 the SS troops took over the internal affairs of Hungary and proceeded to organize the deportation of the Jews. To the Nazis this was a routine assignment; within hours all local officials were informed of operational plans. The high command issued a directive designed to placate Jewish fears and induce cooperation. It was announced that the Jews would be shipped to Poland as an emergency labor force and that they were only being drafted for temporary work. There were many who believed this version. Others, less credulous, resigned themselves and hoped for the best. Still others began to make plans for escape. By April 13 the Hungarian Jews were being rounded up from all over the country in what was once a huge brick factory. The rest is well known.

I was working in Ungvar and usually came home on weekends. At that time it was no longer possible for a Jew to travel freely.



```
]: print (ocr_result)

"GABRIEL Meamall

On Easter movning in the year 1944, I took my six-year-old
son by the hand and began walking fron my home town toward the
valleys and forests of the Carpathizn mountains. For nearly
eight months we lived in barns, attics and makeshift eabins. With
the gene nous help of an unusually courageous man, we managed to
survive Europe's greatest fit of madness. Those who walked in
the opposite direction on that Easter day were lese fortunate.
They were taken in trainloads to places whose once obscure names
are now, and forever will be, synonymous with terror, evil and
death. What follows is our story of survival told to the best
of my ability, in plain, simple language.

In March of 1944 the SS troops took over the internal affairs
of Hungary and proceeded to organize the deportation of the dows.
To the Nazie thie was a routine assignment; within hours all local
officials were informed of operational plans. The high command
issued a directive designed to placate Jewish fears and induce
cooperation. It was announced that the Jews would be shipped to
Poland as an emergency labor force and that they were only being
drafted for temporary work. There were many who believed this
version. Others, less credulous, resigred themselves and hoped for
the vest. Still others began to make plans for escape. By Aprii
13 the Hungarian Jews were being rounded up from all over the .
country in what was once a huge brick factory. The rest is well
known. :

J was working in Ungvar and usually came home on weekends.

At that time it was no longer possible for a Jew to travel freely.
```

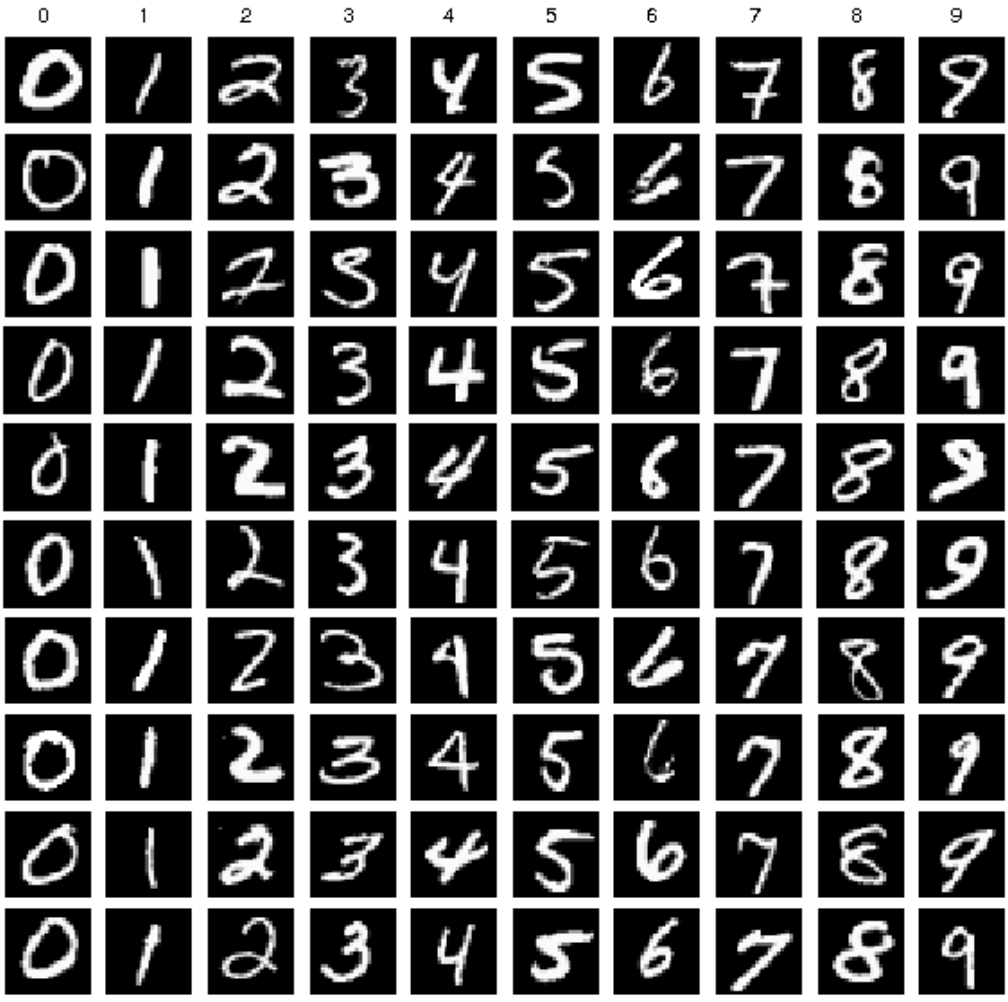
Scanned Image

Text Extracted

I learnt the step-by-step procedure of preprocessing an image so that it is ready for going through OCR by Pytesseract. For this, I have to grayscale the image, followed by deskewing (straightening) it , then reduce noise in the image (noise refers to the graininess in the image) after which I have to use a suitable kernel and apply thresh onto the image, after this we have to find the contours in the image and OCR the contour of our interest to get the text.
The image on the left depicts a scanned image that is rotated at a particular angle, for pytesseract to ocr it well, we have to do all the pre-processing described above which gives the result as shown in the image on the right.

Detecting handwritten numbers from the MNIST dataset

I immersed myself into ML concepts and neural network architectures.
Delved into the MNIST dataset and trained neural networks for character recognition.
Achieved notable accuracy in recognizing handwritten characters.



A representation of the MNIST Dataset

The MNIST is a dataset of labeled handwritten digits from 0-9 as shown above, it consists of 70,000 labeled handwritten 28x28 pixel images of digits. Out of these, 60,000 are for training and 10,000 were used for testing the model. Here I used tensorflow. I used a CNN to train the model and used 10 epochs to get as accurate of a model as possible. This resulted in a 99.14% accuracy as shown in the image below.

```
[11]: # Train the model
      model.fit(x_train, y_train, batch_size=128, epochs=10, validation_data=(x_test, y_test))

      Epoch 1/10
      469/469 [==============================] - 26s 55ms/step - loss: 0.3331 - accuracy: 0.8995 - val_loss: 0.0831 - val_accuracy: 0.9752
      Epoch 2/10
      469/469 [==============================] - 27s 58ms/step - loss: 0.1017 - accuracy: 0.9696 - val_loss: 0.0520 - val_accuracy: 0.9829
      Epoch 3/10
      469/469 [==============================] - 32s 69ms/step - loss: 0.0757 - accuracy: 0.9765 - val_loss: 0.0429 - val_accuracy: 0.9856
      Epoch 4/10
      469/469 [==============================] - 31s 66ms/step - loss: 0.0641 - accuracy: 0.9804 - val_loss: 0.0354 - val_accuracy: 0.9885
      Epoch 5/10
      469/469 [==============================] - 30s 64ms/step - loss: 0.0570 - accuracy: 0.9821 - val_loss: 0.0359 - val_accuracy: 0.9882
      Epoch 6/10
      469/469 [==============================] - 29s 61ms/step - loss: 0.0526 - accuracy: 0.9837 - val_loss: 0.0331 - val_accuracy: 0.9881
      Epoch 7/10
      469/469 [==============================] - 30s 65ms/step - loss: 0.0475 - accuracy: 0.9850 - val_loss: 0.0321 - val_accuracy: 0.9889
      Epoch 8/10
      469/469 [==============================] - 30s 64ms/step - loss: 0.0432 - accuracy: 0.9862 - val_loss: 0.0308 - val_accuracy: 0.9894
      Epoch 9/10
      469/469 [==============================] - 30s 64ms/step - loss: 0.0429 - accuracy: 0.9866 - val_loss: 0.0293 - val_accuracy: 0.9902
      Epoch 10/10
      469/469 [==============================] - 27s 58ms/step - loss: 0.0396 - accuracy: 0.9873 - val_loss: 0.0259 - val_accuracy: 0.9914
[11]: <keras.callbacks.History at 0x1b5612648d0>

[12]: # Evaluate the model
      _, accuracy = model.evaluate(x_test, y_test)
      print("Accuracy:", accuracy)

      313/313 [==============================] - 2s 6ms/step - loss: 0.0259 - accuracy: 0.9914
      Accuracy: 0.9914000034332275
```

Output from the CNN

Later on, in the project I realized that the task can be accomplished much faster and much more accurately if I try to detect the “Aadhar Card Number” box in the given document and mask the Aadhaar number relative to the coordinates of the found box.

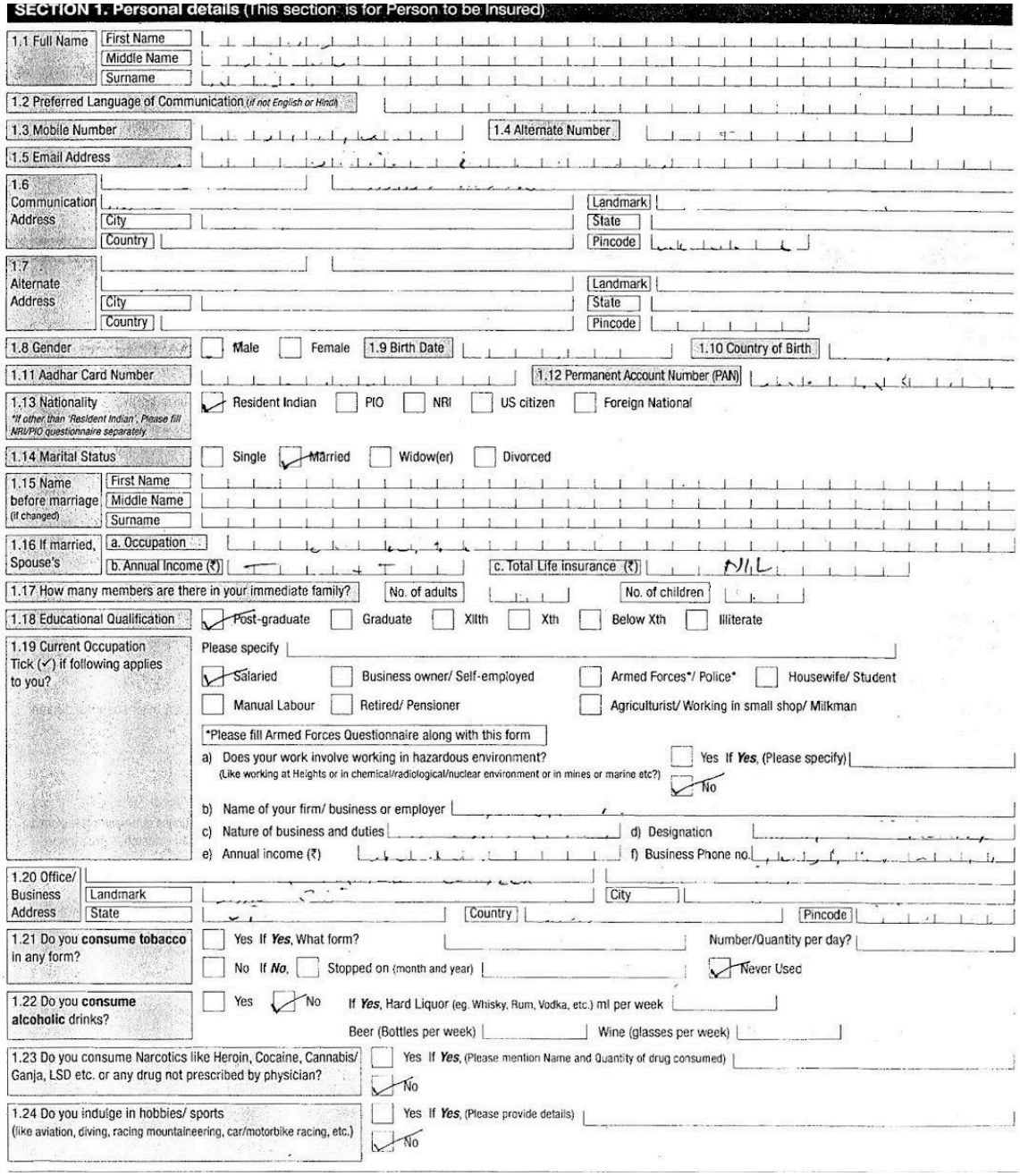
Aadhaar number detection and masking in the document

My first step was to pre-process the image and make it ready for the OCR process. For this I implemented dilation of image (thickening of the characters) followed by erosion(thinning of edges) of the dilated image, followed by another dilation of the now eroded image. What this does is that it makes the edges of all characters and lines very sharp and solid.

After dilation and erosion I use the detect_boxes function to detect the boxes which exist in the document . Within this function, I am using the canny edge detection to detect edges (Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed.)

After getting the edges, I find the contours which would give me the bounding boxes that are present in the document. While I am finding the contour, I specify the maximum width and length range of the bounding box so that only the desired bounding boxes are identified. This

helps in saving time as the algorithm won't waste time in finding the smaller bounding boxes which are a lot in quantity.



SECTION 1: Personal details (This section is for Person to be Insured)

1.1 Full Name — First Name / Middle Name / Surname

1.2 Preferred Language of Communication (if not English or Hindi)

1.3 Mobile Number — 1.4 Alternate Number

1.5 Email Address

1.6 Communication Address — City / Country / Landmark / State / Pincode

1.7 Alternate Address — City / Country / Landmark / State / Pincode

1.8 Gender — ☐ Male ☐ Female — 1.9 Birth Date — 1.10 Country of Birth

1.11 Aadhar Card Number — 1.12 Permanent Account Number (PAN)

1.13 Nationality (*If other than 'Resident Indian', Please fill NRI/PIO questionnaire separately*) — ☑ Resident Indian ☐ PIO ☐ NRI ☐ US citizen ☐ Foreign National

1.14 Marital Status — ☐ Single ☑ Married ☐ Widow(er) ☐ Divorced

1.15 Name before marriage (if changed) — First Name / Middle Name / Surname

1.16 If married, Spouse's — a. Occupation / b. Annual Income (₹) / c. Total Life insurance (₹) NIL

1.17 How many members are there in your immediate family? — No. of adults — No. of children

1.18 Educational Qualification — ☑ Post-graduate ☐ Graduate ☐ XIIth ☐ Xth ☐ Below Xth ☐ Illiterate

1.19 Current Occupation Tick (✓) if following applies to you? — Please specify

☑ Salaried ☐ Business owner/ Self-employed ☐ Armed Forces*/ Police* ☐ Housewife/ Student
☐ Manual Labour ☐ Retired/ Pensioner ☐ Agriculturist/ Working in small shop/ Milkman

*Please fill Armed Forces Questionnaire along with this form

a) Does your work involve working in hazardous environment? (Like working at Heights or in chemical/radiological/nuclear environment or in mines or marine etc?) ☐ Yes If *Yes*, (Please specify) ☑ No

b) Name of your firm/ business or employer

c) Nature of business and duties — d) Designation

e) Annual income (₹) — f) Business Phone no.

1.20 Office/ Business Address — Landmark / City / State / Country / Pincode

1.21 Do you **consume tobacco** in any form? — ☐ Yes If *Yes*, What form? — Number/Quantity per day? ☐ No If *No*, ☐ Stopped on (month and year) ☑ Never Used

1.22 Do you **consume alcoholic** drinks? — ☐ Yes ☑ No If *Yes*, Hard Liquor (eg. Whisky, Rum, Vodka, etc.) ml per week — Beer (Bottles per week) — Wine (glasses per week)

1.23 Do you consume Narcotics like Heroin, Cocaine, Cannabis/ Ganja, LSD etc. or any drug not prescribed by physician? ☐ Yes If *Yes*, (Please mention Name and Quantity of drug consumed) ☑ No

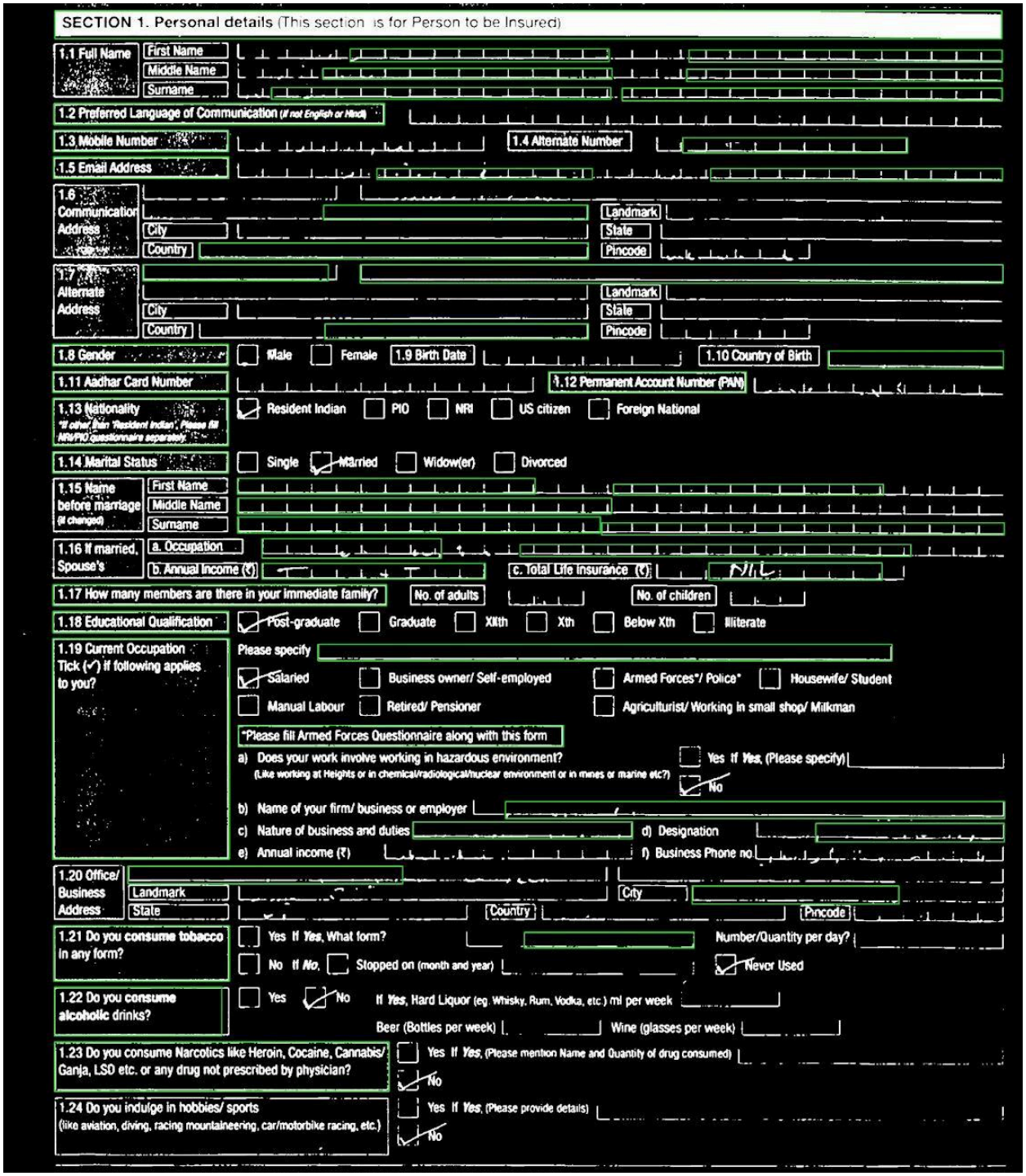
1.24 Do you indulge in hobbies/ sports (like aviation, diving, racing mountaineering, car/motorbike racing, etc.) ☐ Yes If *Yes*, (Please provide details) ☑ No

Application Form Template

Once we identify the correct sizes of bounding boxes, the next step would be to reduce the noise enough so that pytesseract can OCR and find Aadhaar Card or PAN card as text in the document and return a coordinate value of the bounding box from which it detected the text.

For this we use the noise reduction algorithm shown in the OCR in Python : Playlist to obtain a noise-free image which can be used for the OCR. After getting the image without noise, we pass

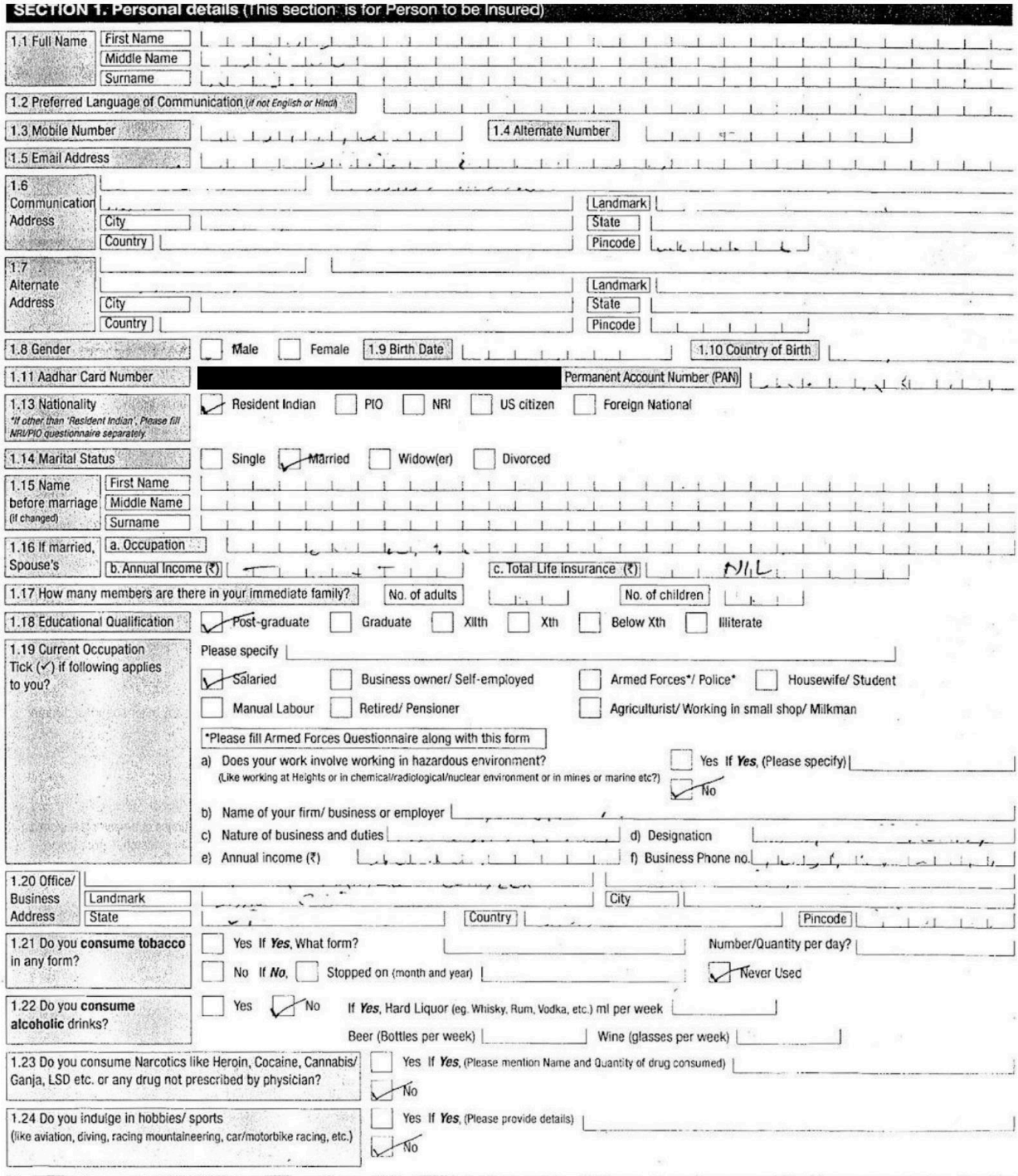
it through pytesseract and specify to OCR only the regions covered by the bounding boxes found earlier.



Bounding Boxes (green) detected within the document

If there is a text match of any of the target words with the bounding box, the location of the box is masked in the original image and the masked image is returned.

This concludes the basics of the entire masking process.



SECTION 1. Personal details (This section is for Person to be Insured)

1.1 Full Name — First Name / Middle Name / Surname

1.2 Preferred Language of Communication (if not English or Hindi)

1.3 Mobile Number — 1.4 Alternate Number

1.5 Email Address

1.6 Communication Address — Landmark / City / State / Country / Pincode

1.7 Alternate Address — Landmark / City / State / Country / Pincode

1.8 Gender — ☐ Male ☐ Female — 1.9 Birth Date — 1.10 Country of Birth

1.11 Aadhar Card Number — [redacted] — Permanent Account Number (PAN)

1.13 Nationality *If other than 'Resident Indian', Please fill NRI/PIO questionnaire separately* — ☑ Resident Indian ☐ PIO ☐ NRI ☐ US citizen ☐ Foreign National

1.14 Marital Status — ☐ Single ☑ Married ☐ Widow(er) ☐ Divorced

1.15 Name before marriage (if changed) — First Name / Middle Name / Surname

1.16 If married, Spouse's — a. Occupation — b. Annual Income (₹) — c. Total Life Insurance (₹) NIL

1.17 How many members are there in your immediate family? — No. of adults — No. of children

1.18 Educational Qualification — ☑ Post-graduate ☐ Graduate ☐ XIIth ☐ Xth ☐ Below Xth ☐ Illiterate

1.19 Current Occupation Tick (✓) if following applies to you? — Please specify

☑ Salaried ☐ Business owner/ Self-employed ☐ Armed Forces*/ Police* ☐ Housewife/ Student

☐ Manual Labour ☐ Retired/ Pensioner ☐ Agriculturist/ Working in small shop/ Milkman

*Please fill Armed Forces Questionnaire along with this form

a) Does your work involve working in hazardous environment? (Like working at Heights or in chemical/radiological/nuclear environment or in mines or marine etc?) ☐ Yes If *Yes*, (Please specify) ☑ No

b) Name of your firm/ business or employer

c) Nature of business and duties — d) Designation

e) Annual income (₹) — f) Business Phone no.

1.20 Office/ Business Address — Landmark / City / State / Country / Pincode

1.21 Do you **consume tobacco** in any form? — ☐ Yes If *Yes*, What form? — Number/Quantity per day? ☐ No If *No*, ☐ Stopped on (month and year) ☑ Never Used

1.22 Do you **consume alcoholic** drinks? — ☐ Yes ☑ No If *Yes*, Hard Liquor (eg. Whisky, Rum, Vodka, etc.) ml per week — Beer (Bottles per week) — Wine (glasses per week)

1.23 Do you consume Narcotics like Heroin, Cocaine, Cannabis/ Ganja, LSD etc. or any drug not prescribed by physician? ☐ Yes If *Yes*, (Please mention Name and Quantity of drug consumed) ☑ No

1.24 Do you indulge in hobbies/ sports (like aviation, diving, racing mountaineering, car/motorbike racing, etc.) ☐ Yes If *Yes*, (Please provide details) ☑ No

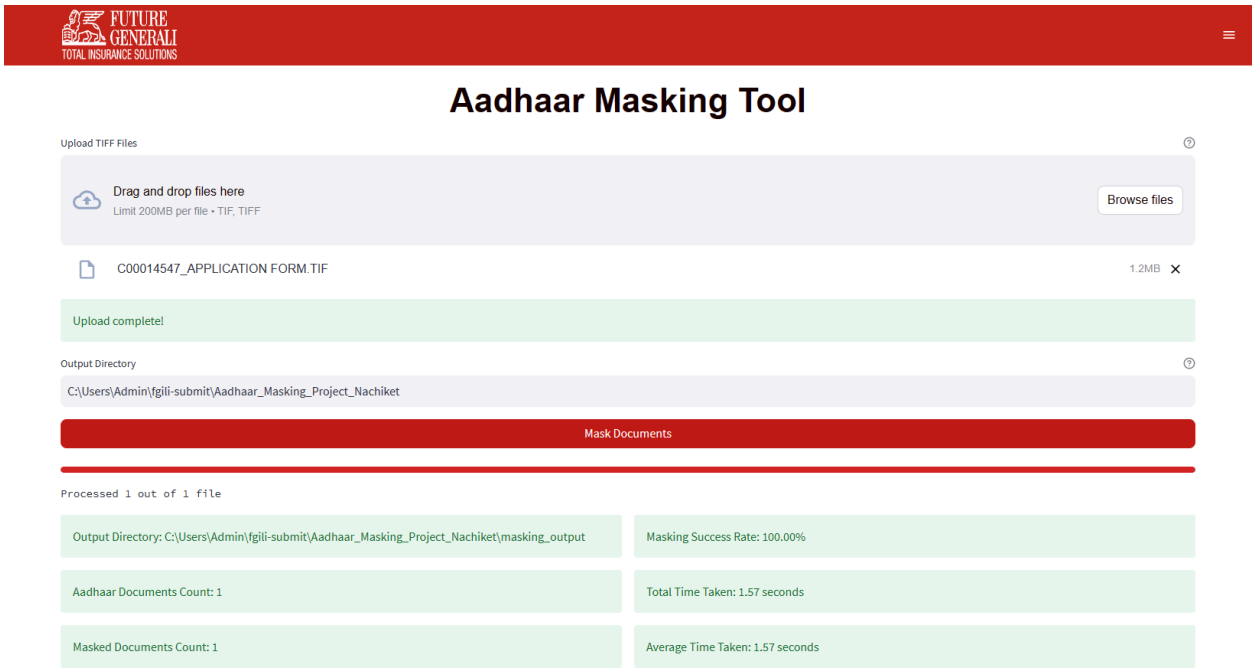
Aadhaar number masked in the document

Creating a UI for this project

The next task that came my way was to create a UI for this application. This was the harder side of the project for me, since I had no experience in integrating UI with my projects. For this I relied on streamlit. I used its functions like st.fileUploader, st.buttons, st.columns, st.input to create an intuitive design which runs well with the look and feel of Future Generali websites.

This includes converting the theme to have red as a primary color, the shade of red was picked using color picker on the other official websites of Future Generali. The navbar was specifically designed to mimic the navbar of their websites.

Using streamlit was a significant struggle for me since it was completely alien for me. This was the first time I was using design extensively. This phase was marked by a lot of back and forth with my mentor, Nikita Malviya, who guided me with feedbacks through the UI design process to best recreate the Future Generali look even with my naive streamlit skills.



The UI of The Aadhaar Masking Tool that I created

Extra: Creation of a UI for their pre-existing solution

To create a UI for their pre-existing solution, I had to work with an API that my mentor, Nikita Malviya, had created. Her Aadhaar masking API was proficient in masking the Aadhaar numbers from the Aadhaar cards and even with scanned documents of different kinds but it did not perform as well with this subset of documents that Future Generali had. For this phase, I undertook the task of reading the API documentation and understanding all the values returned and how to use the POST command. I feel that this knowledge will be helpful in the future as I

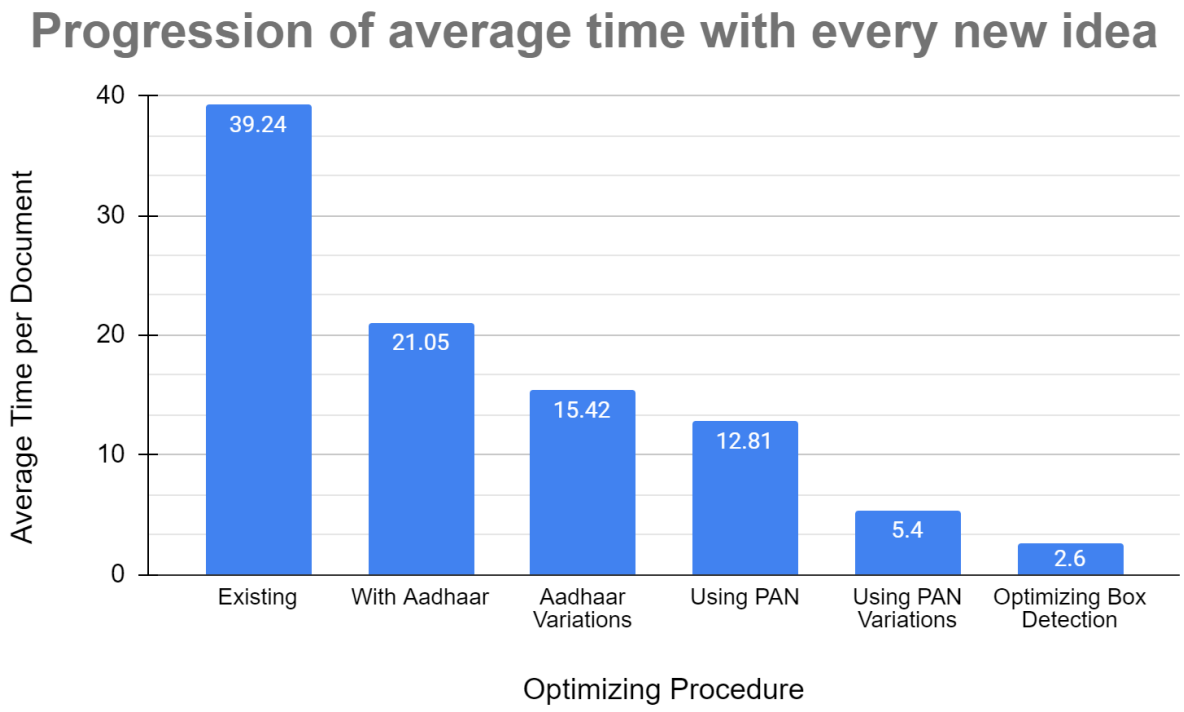
will be working with APIs. The UI for the API was the same as that of my application, but the inner functions are structured a bit differently to suit the API request outputs.

Evolution of Approach : Optimization

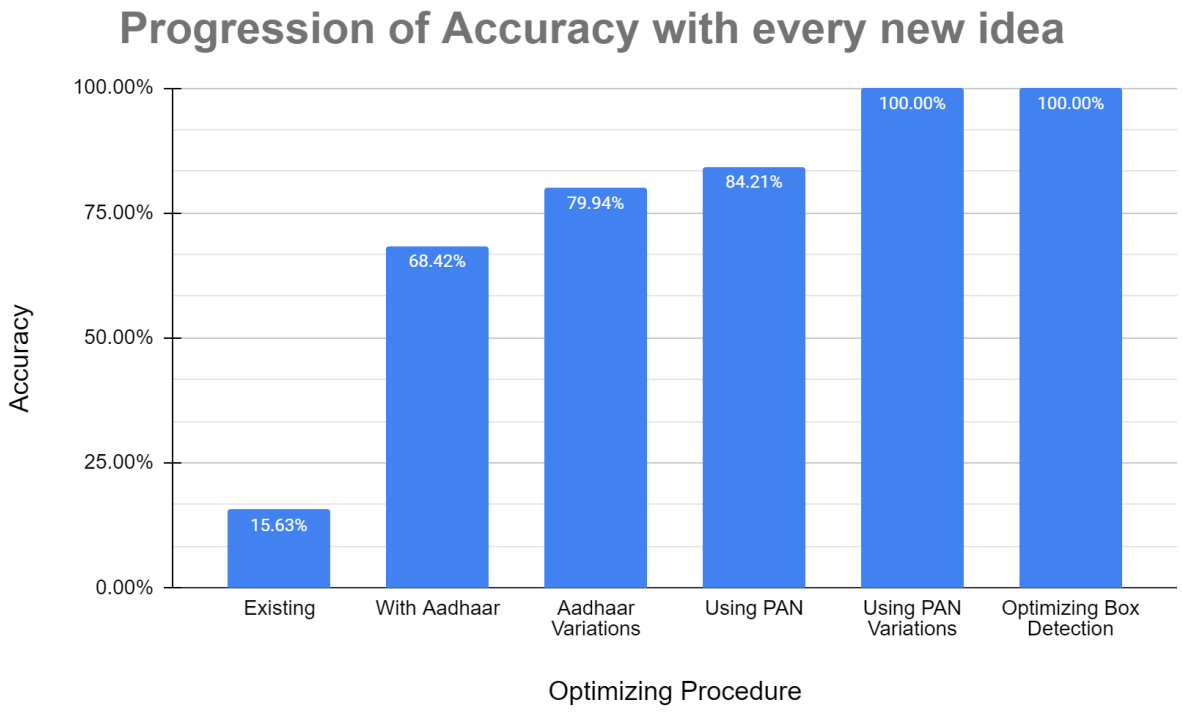
The initial versions of the algorithm worked well with the scanned documents but they weren't 100% accurate with the training set of 19 documents that I had and they were relatively slow, taking about 21 seconds to process a single image. This section covers my approach and optimizations which brought the 21 seconds down to 2.6 seconds per document. The visual representation of progress of the average time and accuracy with every new idea is shown below in the form of graphs.

- With Aadhaar : The masking process will only be successful if the target word matches with the text obtained from performing the OCR on the noise-removed image. In the earliest of the iterations, I was only using the target word “Aadhar” and hence it had a lower detection rate and a slow speed. It took 21.o5 seconds per document to be processed while the accuracy stood at 68.42%.

- Aadhaar Variations : This is when I started looking for alternatives of the target words. For example, I added the target words : “ar Card”, “ Aad”, “har Nu” since the chance of a faulty detection with them was less than one in 100 million. This improved my accuracy as it rose to 79.94% and the time got cut down to 15.42 seconds per document.

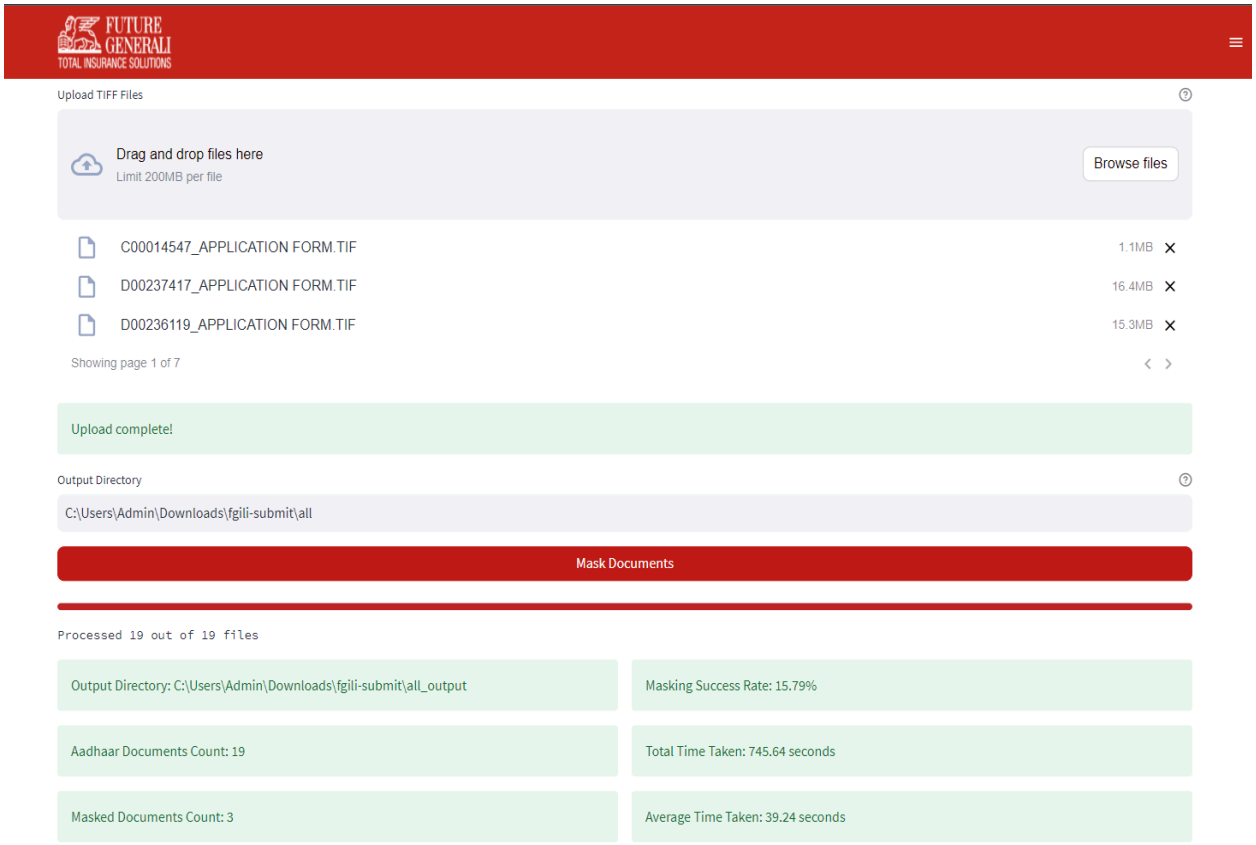


- Using PAN : Till now I was only using the target words which apply to Aadhaar to mask the Aadhaar number, but since I was only dealing with Future Generali applications, I found a pattern that both Aadhaar details and PAN details are always asked simultaneously in the same line in each of the document that exists. Hence I made an algorithm which could detect the PAN details box too and using that information, it would mask the Aadhaar number since the relative position of the Aadhaar number from PAN number was known. This increased the accuracy to 84.21% as more boxes could be discovered and I effectively had two ways of getting the same Aadhaar number masked. With this, even the speed increased to 12.81 seconds per document.
- Using PAN Variations : I then used the same logic as before and created target words like " count", " (PAN" and " Permanent Acco" so that we could bypass the spelling errors that pytesseract does when it performs OCR on the document. Simultaneously, I tweaked the code to only scan the first page of the document as only the first page contains the Aadhaar number. This increased the accuracy to 100% and increased the speed to about 5.4 seconds per document.
- Optimizing Box Detection : Even after this, I wanted to make the application more efficient at detecting the boxes. So I further restricted the search area to the first 2/3rd of the document. Now the application would look for an Aadhar number only in the first 2/3rd which would lead to fewer detected boxes and hence faster OCR. This increased the speed to 2.6 seconds per document and this is where the Aadhaar masking tool speed is at the time of writing this report.



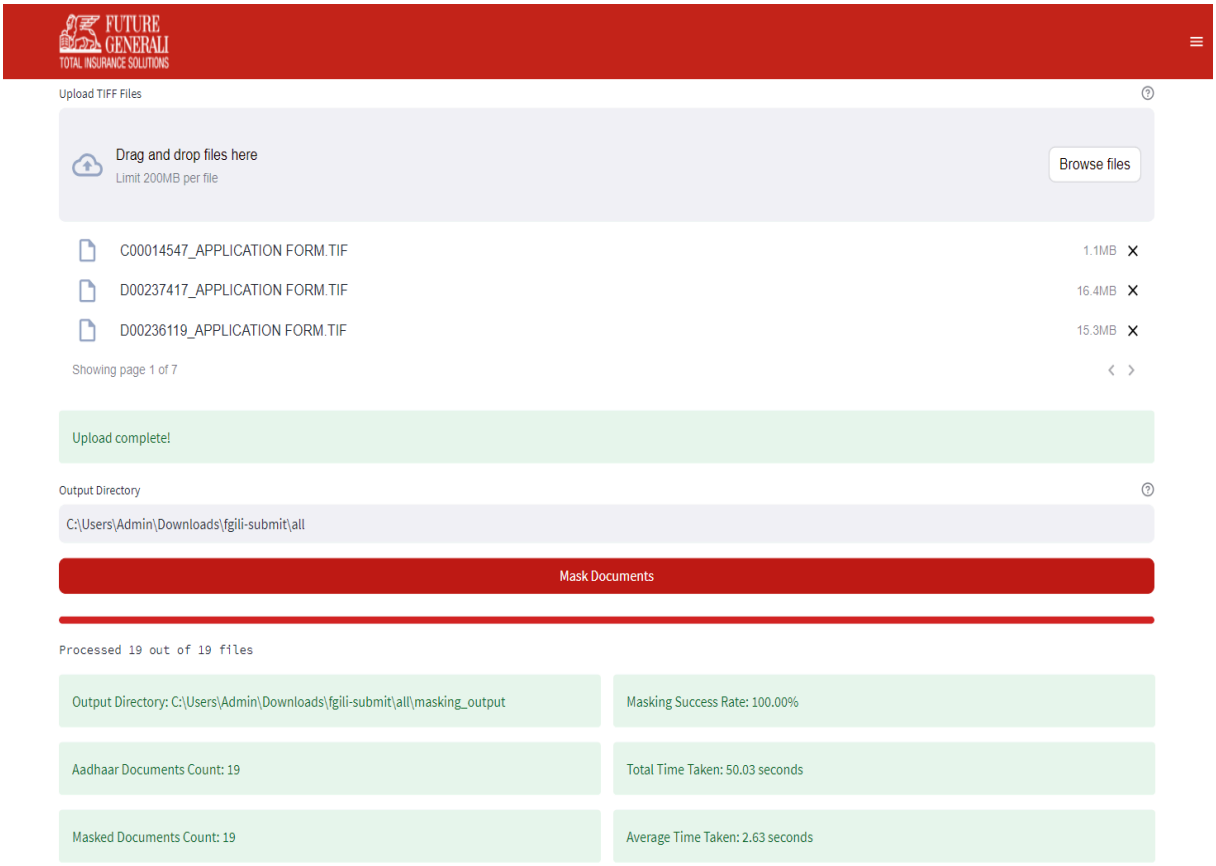
Comparison Of Solutions

This section would provide a brief breakdown of comparison of results between the existing solution and my solution.
Note: The existing solution was very proficient at masking the Aadhaar details from a variety of documents with reasonable accuracy. Meanwhile my Aadhaar masking tool only focuses on a subset of handwritten applications that Future Generali has. Thus my solution would perform poorly with a general document when compared to the existing solution.
This comparison is just based on the results from my tool wherein the application forms on which I had to work with were used.



Using the pre-existing solution

The above image displays the output from the existing solution processing the 19 handwritten applications. Since it is of a generalized nature it could only mask 3 of the 19 documents and hence the accuracy of 15.79%. Furthermore it took 39.24 seconds on average to process every document.



Results from my solution

The image above showcases the output from my solution. The same 19 application forms were uploaded as before. My Aadhaar masking tool processed all the documents with a masking accuracy of 100% and took 2.63 seconds on average to process a particular document. This is 15 times faster than the existing solution.

CONCLUSION

Accomplishments and Learnings

This was my first project which went through a development and testing phase, and which will eventually be deployed as well as integrated into pre-existing solutions for actual users. The initial objective of the project was just to create an Aadhaar masking code which could mask the Aadhaar number as well as the pre-existing solution but as the accuracy of the code improved and eventually superseded the pre-existing code, the project objectives also grew and new tasks emerged. I had to now design a user interface for the pre-existing API.

Future Work

The core learnings from this project can be taken further into projects like:

1. Data Entry Automation application which can automate data entry tasks to save time and reduce errors in the process.
2. E-commerce and Retail where my image processing and OCR techniques could be used to extract product information from images, classify products or automate inventory management
3. Natural Language Processing (NLP) wherein I can work on extracting information from scanned documents and perform advanced language processing tasks like sentiment analysis or topic modeling.
4. Making an API for Aadhaar Masking Tool which can be of a more generalized form but with the ability to be customized as per the organization's requirements.

Self - Review

As this report comes to a close, I would like to reminisce on the project and write a few things that I would have done differently if I started off with the knowledge that I have now. These things are:

1. Incorporating Parallel Processing : This has been on the back of my mind for the last few days since completing the project that the entire program can be much faster if I include the parallel processing capabilities. I did not implement it in this project primarily due to not knowing much about parallel processing or multi-processing. This is something that I will look forward to incorporating in my future projects.
2. Better File Management : I admit that I haven't been the best at keeping track of my program files and the different variations that I tried and tested during the ideation process. Having better file management would enable me to find what I want faster and would help in the tool development process.

REFERENCES

1. OCR in Python : Playlist

2. Handwritten Digit Recognition

3. Streamlit Documentation

4. Stack Overflow